Computational Topology Project Topological Encoding of Paths in the Task Space

Abdullah Akce

aakce2@illinois.edu

I. Preword

The proposed work lies in the intersection of robotics and topology. Our problem statement is not an open problem in topology, but it proposes the use of topological tools to solve a problem in robotics in the most efficient way.

II. Motivation

We are interested in designing brain-machine interfaces. These interfaces are controlled by a human user through a noisy and low-bandwidth neural sensor which measures brain activity. Our particular interest is to enable users of these interfaces to control a robotics system (e.g. a wheelchair). Our recent work demonstrated that we can learn an approximate of the user's intended path (2-dimensional path in the obstacle-free plane) by using binary inputs obtained from an electroencephalography (EEG) sensor. In EEG-based brain-machine interfaces, the input is usually one or two bits (with noise) at a rate of about one second. Because of this limitation, it takes quite a long time to specify a path. Instead of specifying the path itself with this limited input source, we can consider specifying the type of it and let the system figure out a good representative path for that type. In the context of this paper, the type of a path may correspond to different things. For example, we can consider the homotopy class of a path as its type.

III. Problem Statement

In motion planning, the fundamental problem is to find a feasible path between a starting and a goal point in the configuration space of the robot. Combinatorial methods solve this problem by first computing a cell decomposition of the configuration space, then generating a graph to capture the incidence relation of cells and finally solving the associated graph search problem. When the configuration space and the starting point are known, the input for a motion planner is just the goal point. The motion planner can then compute a feasible path, if one exists. We are interested in a different motion planning problem. Our planner knows the task space and the starting point in the task space but the goal is not known. Instead of being fully autonomous and generating a feasible path between two known points, our planner should generate a path from binary commands received from a sensor. We also require that each binary sequence corresponds to a path in the task space. Moreover receiving a command is very expensive. So, in this work we are interested in encoding of paths using minimum number of bits. An arbitrary path in the task space $\mathbb{T}$ can be represented by $\sigma : [0,1] \to \mathbb{T}$. This function cannot be exactly encoded by a binary sequence, instead methods rely on computing an approximation of it.

In this work, our approach will be to approximate paths with an equivalence class defined by the topology and/or the geometry of the task space. We consider two paths from the same equivalence class as being equally good representatives of that class. If the task space is a 2-manifold and if the goal is known, we can define equivalence classes as homotopy classes of paths. Our problem is then to represent an equivalence class with minimum number of bits. In Section IV, we consider this simple case and give some preliminary results. Our work will build upon this preliminary work and establish minimal representations for equivalence classes defined in higher dimensional task spaces. For example, consider the manipulation problem for a humanoid robot in the three dimensional space shown in Figure 1. Path-homotopy in this space is not a good candidate for constructing equivalent classes. The problem with path-homotopy in three or higher dimensions is that a path can be deformed into a very large number of paths by using the extra degree of freedom — a consequence which does not match with how humans perceive and compare objects in the real-world. It suggests that we should define a metric between homotopic paths and divide homotopy classes further into subclasses such that the maximum distance between any two paths of a subclass is bounded. In order to compute distances between paths, we can use the Frechet disntace [1] or the height of the homotopy [2].

IV. Preliminary work

In this section, we demonstrate our work towards a minimal representation of homotopy classes in the plane with holes, in terms of a crossing sequence of a cell decomposition. We further require that the cell decomposition preserves the connectivity of the free space. There are a variety of cell decomposition methods that have been proposed in the robotics community (see [4], [5]) but the author is not aware of any approaches that give a minimal decomposition for our problem. Here, we will consider triangular decompositions.

In Figure 2, we show how to walk in a triangulation with binary commands. This idea has been used in surface encoding methods to compress the connectivity of a surface. (see [6], [7]). In Figure 3, we show that we can do better by ignoring the shape of obstacles and doing the triangulation more carefully.

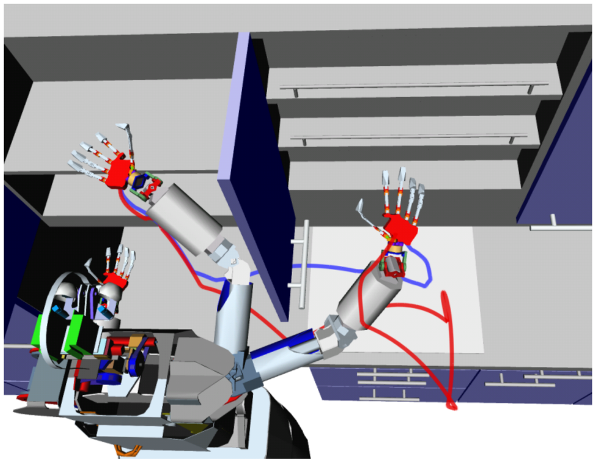

Fig. 1. This figure appears in [3]. It is a good example for illustrating how we can extend our proposed work to higher dimensions. Here, the robotics system is a humanoid in a human-centered environment. The figure shows two paths (red and blue) for the right hand of the humanoid. In the context of our work, we say that these two paths have the same type.

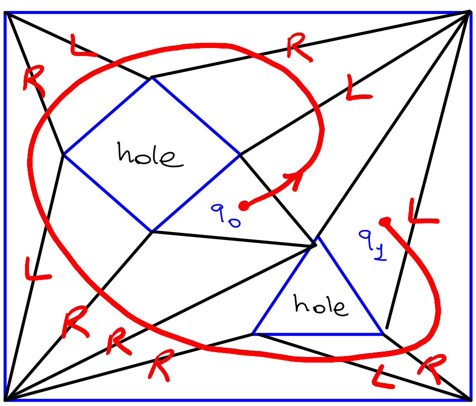

Fig. 2. A triangulation in the plane with polygonal holes. Blue edges represent the boundary. Black edges are the diagonals of the triangulation. We can walk in this triangulation by using just two commands: left (L) and right (R), provided that the starting configuration and the initial edge crossing are given. The path (shown in red) from the starting configuration q_0 to the goal q_1 can be represented by the crossing sequence: LRLRLRRRLRL in just 11 bits.

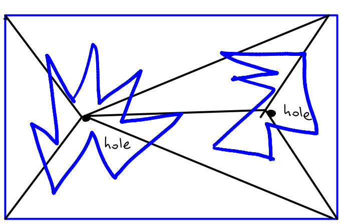

Fig. 3. A triangulation using only one sentinel point for each hole. A crossing sequence of moves in such a triangulation is enough for determining the homotopy type. Using vertices of the interior polygon can result in a much more complex triangulation and would require longer sequences to represent homotopy classes.

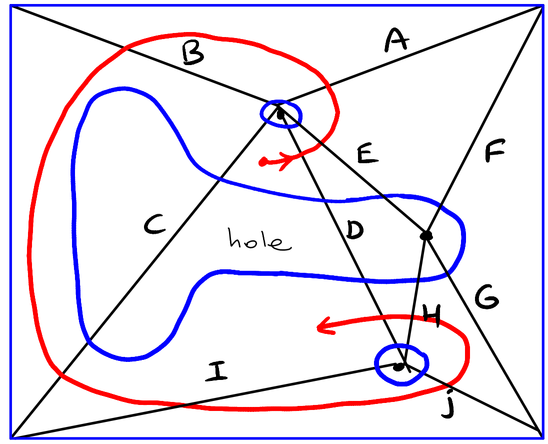

Fig. 4. An example where triangulation with a single sentinel point for each hole does not preserve connetivity of the task space. For example, you cannot cross the edge E after crossing the edge H.

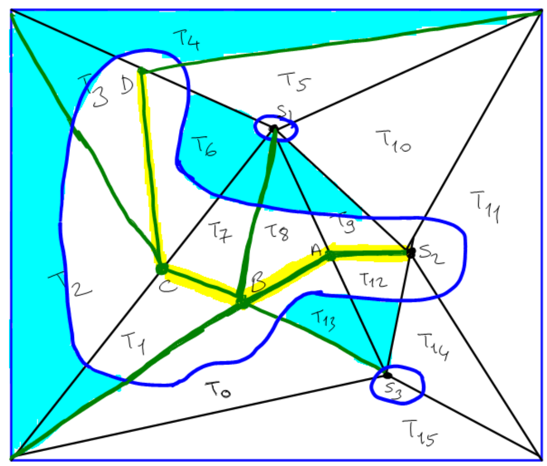

Fig. 5. A triangulation where we preserve the topology. Blue edges represent the boundary. Black and green edges represent the diagonal. Edges highlighted with yellow are marked as boundary edges due to the existence of holes. In order to construct such a triangulation, we start by selecting a single sentinel point for each hole. Whenever the connectivity between two edges, say A and B, of a triangle is not satisfied, we introduce a new vertex v on the opposite edge of that triangle and divide the two triangles facing this edge into two smaller triangles. The diagonal from v to the vertex incident to A and B is marked as a boundary edge. Existence of boundary edges allow us to further reduce the number of cells in the decomposition. For example, the triangles T_1, T_2, T_3 and T_4 can be combined into a single cell, since there is only one valid sequence of binary commands from T_1 to T_4 and from T_4 to T_1.

In Figure 4, we show that this triangulation might not preserve the connectivity. Finally, in Figure 5, we demonstrate how to construct a triangulation that preserves the topology.

References

[1] E. W. Chambers, E. Colin de Verdière, J. Erickson, S. Lazard, F. Lazarus, and S. Thite, "Walking your dog in the woods in polynomial time," in *SCG '08: Proceedings of the twenty-fourth annual symposium on Computational geometry*. New York, NY, USA: ACM, 2008, pp. 101–109. 1

[2] E. W. Chambers and D. Letscher, "On the height of a homotopy," in *21st Canadian Conference on Computational Geometry*, 2009. 1

[3] T. Asfour, P. Azad, N. Vahrenkamp, K. Regenstein, A. Bierbaum, K. Welke, J. Schröder, and R. Dillmann, "Toward humanoid manipulation in human-centred environments," *Robot. Auton. Syst.*, vol. 56, no. 1, pp. 54–65, 2008. 2

[4] S. M. Lavalle, *Planning Algorithms*. Cambridge University Press, 2006. 1

[5] H. Choset, K. Lynch, S. Hutchinson, G. Kanto, W. Burgard, L. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005. 1

[6] G. Taubin and J. Rossignac, "Geometric compression through topological surgery," *ACM Transactions on Graphics*, vol. 17, pp. 84–115, 1996. 1

[7] J. Rossignac, "Edgebreaker: connectivity compression for triangle meshes," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 5, no. 1, pp. 47–61, Jan-Mar 1999. 1